HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200311047-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Ludmila Cherkasova	Confirmation No.:	7467
Application No.:	10/601,357	Examiner:	K.R. Coulter
Filing Date:	June 23, 2003	Group Art Unit:	2141

Title: SYSTEM AND METHOD FOR MANAGING A SHARED STREAMING MEDIA SERVICE

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 9/28/2007.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $450	☐ 3rd Month $1020	☐ 4th Month $1590

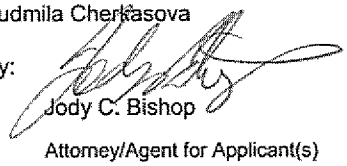
☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,

Ludmila Cherkasova

By: 

Jody C. Bishop

Attorney/Agent for Applicant(s)

Reg No. : 44,034

Date : September 28, 2007

Telephone : (214) 855-8007

I hereby certify that this document is being transmitted to the Patent and Trademark Office via eletronic filing.

Date of Transmission: September 28, 2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 200311047-1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Ludmila Cherkasova et al.

Application No.: 10/601,357 Confirmation No.: 7467

Filed: June 23, 2003 Art Unit: 2141

For: SYSTEM AND METHOD FOR MANAGING A SHARED STREAMING MEDIA SERVICE Examiner: K. R. Coulter

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This Appeal Brief is filed concurrently with a Notice of Appeal and in response to the Final Office Action mailed June 29, 2007.

The fees required under 37 C.F.R. § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Limited Partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

Appellant respectfully notes that the following co-pending application is on appeal before the Board, which contains at least some issues that are similar to issues of the present application, which may be affected or have a bearing on the Board's decision in this appeal. An Appeal Brief was filed for Application No. 10/601,992 on July 19, 2007. No decision has been rendered by the Board as of yet on this appeal.

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 35 claims pending in application.

B. Current Status of Claims

1. Claims canceled: 5, 15, 17, 22
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-4, 6-14, 16, 18-21, 23-39
4. Claims allowed: None
5. Claims rejected: 1-4, 6-14, 16, 18-21, 23-39

C. Claims On Appeal

The claims on appeal are claims 1-4, 6-14, 16, 18-21, and 23-39

IV. STATUS OF AMENDMENTS

A Final Office Action rejecting the claims of the present application was mailed June 29, 2007. In response, Applicant did not file an Amendment in response to the Final Office Action, but instead filed a Notice of Appeal, which this brief supports. Accordingly, the claims on appeal are those as rejected in the Final Office Action of June 29, 2007. A complete listing of the claims is provided in the Claims Appendix hereto.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the separately argued claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. It should be noted that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

According to one claimed embodiment, such as that of independent claim 1, a method for managing admission of requests (e.g., requests 107, 109, 111 of FIG. 1) to a shared media server (e.g., server 101 of FIG. 1) comprises allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective streaming files to clients, wherein said set of shared resources comprises memory (*see* paragraphs 0010 and 0027-0032 of the specification). The method further comprises managing admission of client requests for streaming files to each of the plurality of hosting services to ensure that a desired amount of usage of the shared resources is available to each hosting service (*see* paragraphs 0010 and 0027-0032 of the specification). The managing admission of client requests for streaming files comprises receiving (e.g., operational block 601 of FIG. 6) a client request for a streaming file to be served from one of said hosting services (*see* paragraph 0114 of the specification); and using a segment-based memory model to determine whether at least a portion of the requested streaming file is in the memory (e.g., operational block 602 of FIG. 6, and *see* paragraphs 0031-0036, 0082-0096, and 0114 of the specification).

In certain embodiments, such as that of dependent claim 6, the method further comprises determining from the segment-based memory model a cost associated with the one of said hosting services serving the requested streaming file (e.g., operational block 603 of FIG. 6, and *see* paragraphs 0031-0036 and 0014 of the specification).

In certain embodiments, such as that of dependent claim 8, the managing admission of client requests for streaming files further comprises performing a performance isolation guarantee check for the plurality of hosting services to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

According to another claimed embodiment, such as that of independent claim 11, a system comprises a media server (e.g., server 101 of FIG. 1) comprising a plurality of hosting services for streaming files (e.g., files 102 of FIG. 1) implemented thereon, wherein the media server comprises shared resources and wherein the plurality of hosting services share usage of the media server's shared resources in serving streaming files to their respective clients (e.g., clients 104, 105, and 106 of FIG. 1). The system further comprises an admission controller (e.g., management logic 901 of FIG. 9) for managing admission of client requests (e.g., requests 107, 109, 111 of FIG. 1) for service to each of the plurality of hosting services to ensure that no one of the plurality of hosting services overtakes usage of an undesirably high proportion of the shared resources (*see* paragraphs 0011 and 0027-0032 of the specification). The admission controller is operable to receive (e.g., operational block 601 of FIG. 6) a new request for service of a streaming file by one of the plurality of hosting services, and determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request (e.g., operational block 604 of FIG. 6, and *see* paragraphs 0113-0115 of the specification). And, the admission controller is further operable to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

In certain embodiments, such as that of dependent claim 14, the admission controller is operable to use a segment-based model of the memory resources to determine whether at least a portion of a requested streaming file is in the memory resources, *see* paragraphs 0031-0036, 0082-0096, and 0114 of the specification.

According to another claimed embodiment, such as that of independent claim 18, a method for managing admission of requests (e.g., requests 107, 109, 111 of FIG. 1) to hosting services that share resources comprises allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files (e.g., files 102 of FIG. 1) to clients (e.g., clients 104, 105, and 106 of FIG. 1) thereof, wherein said set of shared resources comprises memory (*see* paragraphs 0012 and 0027-0032 of the specification). The method further comprises, for each of the plurality of hosting services, identifying a desired amount of usage of the set of shared resources to be available for the hosting service (*see* paragraphs 0027-0032 of the specification). The method further comprises isolating usage of the set of shared resources by the plurality of hosting services to ensure that the respective desired amount of usage of the set of shared resources is available to each hosting service. The isolating usage of the set of shared resources comprises specifying, for each of the hosting services, an amount of usage of the set of shared resources to be available, at any time, to the hosting service; and determining whether acceptance of a new request for service by a hosting service will violate, at any point in the future, availability of a specified amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

According to another claimed embodiment, such as that of independent claim 23, a method for managing admission of requests to a hosting service comprises allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files (e.g., files 102 of FIG. 1) to clients (e.g., clients 104, 105, and 106 of FIG. 1) thereof (*see* paragraphs 0013 and 0027-0032 of the specification). The method further comprises, for each of the hosting services, identifying a desired amount of usage of the set of shared resources to be available for the hosting service (*see* paragraphs 0027-0032 of the specification). The method further comprises receiving (e.g., operational block 601 of FIG. 6) a new request for a streaming file to be served by one of the hosting services, and determining, based at least in part on a modeled memory state of the shared resources, a cost to the one of the hosting services for serving the requested streaming file, wherein the cost corresponds to the shared resources to be consumed in serving the requested streaming file (e.g., operational block 603 of FIG. 6, and *see*

paragraphs 0031-0036 and 0014 of the specification). The method further comprises determining, based at least in part on the cost, whether to admit the new request for service by the one of the hosting services (e.g., operational blocks 606-607 of FIG. 6).

In certain embodiments, such as that of dependent claim 27, the determining whether to admit the new request comprises determining whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request (e.g., operational block 604 of FIG. 6, and *see* paragraphs 0113-0115 of the specification); and determining whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

In certain embodiments, such as that of dependent claim 39, the determining said cost, based at least in part on a modeled memory state of the shared resources, comprises determining, based at least in part on a segment-based model of memory of the shared resources (e.g., operational block 602 of FIG. 6, and *see* paragraphs 0031-0036, 0082-0096, and 0114 of the specification).

According to another claimed embodiment, such as that of independent claim 28, a method comprises allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files (e.g., files 102 of FIG. 1) to clients (e.g., clients 104, 105, and 106 of FIG. 1) thereof, wherein the shared resources includes a memory (*see* paragraphs 0013 and 0027-0032 of the specification). The method further comprises receiving, at a time T_{cur}, a new request for a streaming file to be served by one of the hosting services (e.g., operational block 601 of FIG. 6, and *see* paragraphs 0113-0114), and creating a segment-based model of the memory as of time T_{cur} (e.g., operational block 602 of FIG. 6, and *see* paragraphs 0031-0036, 0082-0096, and 0114 of the specification). The method further comprises, based at least in part on the segment-based model of the memory, determining whether to accept the received request for service by the hosting service (e.g., operational blocks 606-607 of FIG. 6).

In certain embodiments, such as that of dependent claim 30, the determining whether to accept the received request for service by the hosting service comprises determining whether the requested hosting service has sufficient available resource usage allocated thereto to service the received request (e.g., operational block 604 of FIG. 6, and *see* paragraphs 0113-0115 of the specification); and determining whether acceptance of the received request for service by the requested hosting service will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

In certain embodiments, such as that of dependent claim 31, the segment-based model of the memory comprises (a) identification of unique segments of streaming files previously accessed by clients and (b) identification of corresponding timestamps of most recent accesses of each unique segment (*see* paragraphs 0034-0036 and paragraphs 0066-0069 of the specification).

According to another claimed embodiment, such as that of independent claim 32, software code stored to a computer-readable medium is provided, which when executed causes a computer to perform a method comprising creating a segment-based model of a media server's memory (*see* paragraphs 0015, 0031-0036, and 0082-0096 of the specification), wherein the media server's memory is a shared resource to which a plurality of hosting services implemented on the media server (e.g., server 101 of FIG. 1) have access for serving their respective files (e.g., files 102 of FIG. 1) to clients (e.g., clients 104, 105, and 106 of FIG. 1) thereof; and determining whether to serve a requested streaming file from one of the plurality of hosting services based at least in part on the segment-based model of the media server's memory (*see* paragraph 0015 of the specification, and operational blocks 606-607 of FIG. 6).

In certain embodiments, such as that of dependent claim 33, the code for determining whether to serve a requested streaming file from one of the plurality of hosting services comprises code, which when executed causes a computer to perform a method comprising determining whether the one of the plurality of hosting services has sufficient available resource usage allocated thereto to serve the requested streaming file (e.g., operational block 604 of FIG. 6, and *see* paragraphs 0113-0115 of the specification); and determining whether acceptance of

the received request for service by the one of the plurality of hosting services will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

In certain embodiments, such as that of dependent claim 34, the segment-based model of the media server's memory comprises (a) identification of unique segments of streaming files previously accessed by clients of the media server and (b) identification of corresponding timestamps of most recent accesses of each unique segment (*see* paragraphs 0034-0036 and paragraphs 0066-0069 of the specification).

According to another claimed embodiment, such as that of independent claim 36, an admission controller (e.g., management logic 901 of FIG. 9) for managing admission of requests to hosting services that share resources comprises means (e.g., an admission controller and/or software stored to computer-readable medium and being executed by a computer, *see e.g.,* paragraphs 0016 and 0144-0151 of the specification) for receiving (e.g., operational block 601 of FIG. 6) a new request for a streaming file (e.g., files 102 of FIG. 1) to be served by one of a plurality of hosting services that share access to a set of shared resources for serving their respective files to clients (e.g., clients 104, 105, and 106 of FIG. 1) thereof. The admission controller further comprises means (e.g., an admission controller and/or software stored to computer-readable medium and being executed by a computer, *see e.g.,* paragraphs 0016 and 0144-0151 of the specification) for performing a resource availability check for the one of a plurality of hosting services from which the streaming file is requested by the new request to determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request (e.g., operational block 604 of FIG. 6, and *see* paragraphs 0113-0115 of the specification). The admission controller further comprises means (e.g., an admission controller and/or software stored to computer-readable medium and being executed by a computer, *see e.g.,* paragraphs 0016 and 0144-0151 of the specification) for performing performance isolation guarantee check for the plurality of hosting services to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired

amount of usage of the shared resources for any of the plurality of hosting services (e.g., operational block 605 of FIG. 6, and *see* paragraphs 0113-0140 of the specification).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 32-35 are rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter.

B. Claims 1-4,6-14,16,18-21, and 23-39 are rejected under 35 U.S.C. §102(e) as being anticipated by published U.S. Patent Application No. 2002/0152305 to Jackson et al. (hereinafter "*Jackson*").

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board reverse the outstanding rejections in light of the remarks contained herein. The claims do not stand or fall together. Instead, Appellant presents separate arguments for various claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

A. Rejections Under 35 U.S.C. § 101

Claims 32-35 are rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter, *see* page 2 of the Final Office Action. The Final Office Action asserts that claims 32-35 are directed to "software that is not implemented on a computer-readable storage medium". *Id.* The Final Office Action further asserts that data structures not claimed as embodied in computer-readable media are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer. *Id.*

However, independent claim 32 recites:

> Software code stored to a computer-readable medium, which when executed causes a computer to perform a method comprising:
> creating a segment-based model of a media server's memory, wherein the media server's memory is a shared resource to which a plurality of hosting services implemented on the media server have access for serving their respective files to clients thereof; and
> determining whether to serve a requested streaming file from one of the plurality of hosting services based at least in part on the segment-based model of the media server's memory. (Emphasis added).

Thus, the Examiner's assertion in the Final Office Action is unsupported and incorrectly characterizes claims 32-35. Appellant respectfully submits that the claims are directed to proper statutory subject matter under 35 U.S.C. §101. For example, M.P.E.P. §2106 explains that "When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized."

In addition, the claim is not directed to a data structure that comprises descriptive material per se. Instead, the claim is directed to software that "when executed causes a computer to perform a method comprising: creating a segment-based model of a media server's memory … and determining whether to serve a requested streaming file from one of the plurality of hosting services…." Thus, claim 1 is clearly directed to functional material.

Accordingly, claims 32-35 are directed to proper statutory subject matter, and therefore Appellant requests that this rejection of these claims be overturned.

B. Rejections Under 35 U.S.C. § 102(b) Over *Jackson*

Claims 1-4,6-14,16,18-21, and 23-39 are rejected under 35 U.S.C. §102(e) as being anticipated by *Jackson*. Appellant respectfully traverses these rejections below.

In order to anticipate a claim under 35 U.S.C. § 102, a reference must teach every element of the claim. *See* M.P.E.P. § 2131. *Jackson* does not teach every element of claims 1-4,6-14,16,18-21, and 23-39, as discussed below, and therefore the rejection of these claims should be overturned.

Independent Claim 1 and Dependent Claims 2-4, 7, and 9-10

Independent claim 1 recites:

> A method for managing admission of requests to a shared media server, the method comprising:
> allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective streaming files to clients, wherein said set of shared resources comprises memory; and
> managing admission of client requests for streaming files to each of the plurality of hosting services to ensure that a desired amount of usage of the shared resources is available to each hosting service, wherein said managing admission of client requests for streaming files comprises:
> receiving a client request for a streaming file to be served from one of said hosting services; and
> using a segment-based memory model to determine whether at least a portion of the requested streaming file is in the memory. (Emphasis added).

Jackson fails to teach at least the above-emphasized element of claim 1. That is, *Jackson* fails to teach using a segment-based memory model to determine whether at least a portion of a requested streaming file is in the memory of a shared media server.

In general, it appears that *Jackson* proposes a content delivery system in which a server is implemented with a plurality of independent engines (such as engines 1030-1070 of *Jackson's* Fig. 1) to perform independent tasks associated with serving requests, *see e.g.*, paragraphs 0085-0086 and 0166 of *Jackson*. In this manner, *Jackson* teaches that the various tasks can be

performed by the independent resources of each engine, and a given engine can thus be upgraded if desired to increase its respective capacity for performing its respective tasks. Thus, the resources that are available for performing different tasks associated with serving a client request may be segregated into different engines.

Jackson appears to mention at paragraph 0133 thereof that multiple web sites may be hosted by a server, and bandwidth limits for each web site may be monitored in order to reject additional requests made to a web site when its respective bandwidth limit is exceeded. In this regard, *Jackson* appears to make some mention of managing admission of requests to a shared media server. However, in managing its admission of client requests, *Jackson* fails to teach a segment-based memory model, or using such a segment-based memory model to determine whether at least a portion of a requested streaming file is in the memory of a shared media server, as recited by claim 1.

The first Office Action (mailed January 16, 2007) asserts (in its treatment of claim 5) that *Jackson* teaches such use of a segment-based memory model at Fig. 3 and paragraphs 0096 and 0107, *see* page 4 of the first Office Action. Paragraphs 0096 and 0107 of *Jackson* merely provide:

> [0096] With regard to the network protocol stack, the stack in traditional systems may often be rather large. Processing the entire stack for every request across the distributed interconnect may significantly impact performance. As described herein, the protocol stack has been segmented or "split" between the network interface engine and the transport processing engine. An abbreviated version of the protocol stack is then provided across the interconnect. By utilizing this functionally split version of the protocol stack, increased bandwidth may be obtained. In this manner the communication and data flow through the content delivery system 1010 may be accelerated. The use of a distributed interconnect (for example a switch fabric) further enhances this acceleration as compared to traditional bus interconnects.
>
> ...
>
> [0107] The embodiment of FIG. 1A contemplates that the protocol processing is shared between the transport processing engine 1050 and the network interface engine 1030. This sharing technique may be called "split

> protocol stack" processing. The division of tasks may be such that higher tasks in the protocol stack are assigned to the transport processor engine. For example, network interface engine 1030 may processes all or some of the TCP/IP protocol stack as well as all protocols lower on the network protocol stack. Another approach could be to assign state modification intensive tasks to the transport processing engine.

The above-cited portions of *Jackson* fail to teach, in any way, a segment-based memory model, or using such a segment-based memory model to determine whether at least a portion of a requested streaming file is in the memory of a shared media server. Instead, the above portions of *Jackson* appear to merely discuss implementing a protocol stack.

In response to the above arguments, the Final Office Action asserts on page 6 thereof that *Jackson* teaches using a segment-based memory model to determine whether at least a portion of the requested streaming file is in the memory, citing to paragraphs 123 and 124 of *Jackson*. Paragraphs 123 and 124 of *Jackson* merely provide:

> [0123] In one embodiment storage management engine 1040 may be a dedicated block-level cache processor capable of block level cache processing in support of thousands of concurrent multiple readers, and direct block data switching to network interface engine 1030. In this regard storage management engine 1040 may utilize a POWER PC 7450 processor in conjunction with ECC memory and a LSI SYMFC929 dual 2 GBaud fibre channel controller for fibre channel interconnect to content sources 1090 and/or 1100 via dual fibre channel arbitrated loop 1092. It will be recognized, however, that other forms of interconnection to storage sources suitable for retrieving content are also possible. Storage management engine 1040 may include hardware and/or software for running the Fibre Channel (FC) protocol, the SCSI (Small Computer Systems Interface) protocol, iSCSI protocol as well as other storage networking protocols.
>
> [0124] Storage management engine 1040 may employ any suitable method for caching data, including simple computational caching algorithms such as random removal (RR), first-in first-out (FIFO), predictive read-ahead, over buffering, etc. algorithms. Other suitable caching algorithms include those that consider one or more factors in the manipulation of content stored within the cache memory, or which employ multi-level ordering, key based ordering or function based calculation for replacement. In one embodiment, storage management engine may implement a layered multiple LRU (LMLRU) algorithm that uses an integrated block/buffer management structure including at least two layers of a configurable number of multiple LRU queues and a two-dimensional

> positioning algorithm for data blocks in the memory to reflect the relative priorities of a data block in the memory in terms of both recency and frequency. Such a caching algorithm is described in further detail in U.S. patent application No. 09/797,198, entitled "Systems and Methods for Management of Memory" by Qiu et. al, the disclosure of which is incorporated herein by reference.

While the above portions of *Jackson* mention that any of various caching techniques may be employed for storing data to a cache, it makes no mention whatsoever of a memory model, much less a segment-based memory model. That is, while the system of *Jackson* may include some type of memory (e.g., the above-described cache), *Jackson* simply provides no teaching whatsoever of modeling the memory, and certainly fails to teach a segment-based memory model as recited in claim 1. Further, the above portion of *Jackson* provides no teaching whatsoever of using such a segment-based memory model to determine whether at least a portion of a requested streaming file is in the memory of a shared media server.

Accordingly, the Examiner has failed to establish a proper prima facie case of anticipation regarding at least this element, as the cited portions of *Jackson* in no way teaches the above-identified element. Further, Appellant respectfully submits that no other portion of *Jackson* teaches this element.

In view of the above, Appellant respectfully requests that the rejection of claim 1 be overturned.

Claims 2-4, 7, and 9-10 each depend either directly or indirectly from independent claim 1, and are thus likewise believed to be allowable at least based on their dependency from claim 1 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 2-4, 7, and 9-10 also be overturned.

Dependent Claim 6

Dependent claim 6 depends from claim 1, and thus inherits all of the limitations of claim 1 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 6 is allowable at least because of its dependence from claim 1 for the reasons discussed above.

Further, claim 6 recites "determining from the segment-based memory model a cost associated with the one of said hosting services serving the requested streaming file." *Jackson* fails to teach this further element of claim 6. As discussed above with claim 1, *Jackson* fails to teach a segment-based memory model. Thus, *Jackson* fails to teach determining, from the segment-based memory model, a cost associated with serving a requested streaming file.

Thus, for this further reason, the rejection of claim 6 should also be overturned.

Dependent Claim 8

Dependent claim 8 depends from claim 7, which depends from claim 1, and thus claim 8 inherits all of the limitations of claim 1 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 8 is allowable at least because of its dependence from claim 1 for the reasons discussed above.

Claim 7 recites "wherein said managing admission of client requests for streaming files comprises: receiving a new request for service of a streaming file by one of the plurality of hosting services; [and] performing a resource availability check for the one of a plurality of hosting services to determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request."

Claim 8 further recites "wherein said managing admission of client requests for streaming files further comprises: performing a performance isolation guarantee check for the plurality of hosting services to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services" (emphasis added). As discussed in further detail below with regard

to claim 11, *Jackson* fails to teach performing such a performance isolation guarantee check to determine whether acceptance of a new request will violate at any point in the future availability of the shared resources. Rather, *Jackson* teaches a technique in which new requests are accepted until a defined bandwidth is exceeded, without any consideration of whether acceptance of a given request will violate availability of a desired amount of usage of shared resources by a hosting service at some point in the future.

Thus, for this further reason, the rejection of claim 8 should also be overturned.

Independent Claim 11 and Dependent Claims 12-13 and 16

Independent claim 11 recites:

> A system comprising:
> a media server comprising a plurality of hosting services for streaming files implemented thereon, wherein the media server comprises shared resources and wherein the plurality of hosting services share usage of the media server's shared resources in serving streaming files to their respective clients; and
> an admission controller for managing admission of client requests for service to each of the plurality of hosting services to ensure that no one of the plurality of hosting services overtakes usage of an undesirably high proportion of the shared resources;
> wherein said admission controller is operable to receive a new request for service of a streaming file by one of the plurality of hosting services, and determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request; and
> wherein said admission controller is further operable to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services. (Emphasis added).

Jackson fails to teach at least the above-emphasized element of claim 11. That is, *Jackson* fails to teach an admission controller that is operable to determine whether acceptance of a new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

As discussed above with claim 1, *Jackson* appears to mention at paragraph 0133 thereof that multiple web sites may be hosted by a server, and bandwidth limits for each web site may be monitored in order to reject additional requests made to a web site when its respective bandwidth limit is exceeded. For instance, paragraph 0133 of *Jackson* provides:

> [0133] For example, a content delivery system may contain data for two web sites. An operator of the content delivery system may guarantee one web site ("the higher quality site") higher performance or bandwidth than the other web site ("the lower quality site"), presumably in exchange for increased compensation from the higher quality site. The network interface processing engine 1030 may be utilized to determine if the bandwidth limits for the lower quality site have been exceeded and reject additional data requests related to the lower quality site. Alternatively, requests related to the lower quality site may be rejected to ensure the guaranteed performance of the higher quality site is achieved. In this manner the requests may be rejected immediately at the interface to the external network and additional resources of the content delivery system need not be utilized. In another example, storage service providers may use the content delivery system to charge content providers based on system bandwidth of downloads (as opposed to the traditional storage area based fees). For billing purposes, the network interface engine may monitor the bandwidth use related to a content provider. The network interface engine may also reject additional requests related to content from a content provider whose bandwidth limits have been exceeded. Again, in this manner the requests may be rejected immediately at the interface to the external network and additional resources of the content delivery system need not be utilized.

In this regard, *Jackson* appears to determine whether a bandwidth limit assigned for a given web site being hosted is exceeded, and if it is exceeded, then an additional request for service by such web site is rejected. While this appears to mention rejecting the acceptance of additional requests once the defined bandwidth limit for a given web site is exceeded, this fails to teach determining whether acceptance of a new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services. Rather, this portion of *Jackson* appears to propose accepting new requests until a defined bandwidth is exceeded, without any consideration of whether acceptance of a given request will violate availability of a desired amount of usage of shared resources by a hosting service at some point in the future.

As discussed in the present application at paragraphs 0113-0140, for example, certain instances may arise in which a request may not immediately cause a violation of availability of a desired amount of usage of shared resources (and thus would be accepted in *Jackson*), but acceptance of such request may lead to such a violation at some point in the future. For instance, paragraph 0116 of the present application explains:

> **[0116]** In operational block 605, the admission controller determines whether admission of the newly received request $r_{i\_new}^{f}$ for service by the shared media server will cause violation of the performance isolation guarantee at any point in the future for any of the plurality of logical hosts implemented on the shared media server. As described further herein, by maintaining the guarantee of performance isolation for each hosting service *s* implemented on the shared media server, a desired QoS can be maintained for each of those hosting services. If admission of the request $r_{i\_new}^{f}$ will cause violation of the guarantee of performance isolation for any of the hosting services *s*, operation advances to block 606 whereat the request $r_{i\_new}^{f}$ is rejected (i.e., not admitted for service by the shared media server). Otherwise, if it is determined that admission of the request $r_{i\_new}^{f}$ will not cause a violation of the guarantee of performance isolation for any of the hosting services at any point in the future, operation advances to block 607 whereat request $r_{i\_new}^{f}$ is admitted for service by the shared media server.

The Final Office Action asserts on page 7 thereof that *Jackson* teaches this element of claim 11, citing to "paragraph 9 'predictability'; paragraph 10 'capacity planning'; paragraph 30 'capacity planning' 'predict future loads'; paragraphs 127, 216, 219, and 269". Appellant respectfully disagrees. While various portions of *Jackson* refer generally to providing "greater predictability in the capability of a network server" (e.g., paragraph 0009) and providing for "capacity planning", these general references to predictability and capacity planning comes no where close to teaching the above element of claim 11 with the specificity required for anticipating the claim element under 35 U.S.C. §102. "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Jackson's general references to predictability and capacity planning fail to teach in complete detail as contained in the claim a determination of whether acceptance of a new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services. For instance, such general references in *Jackson* to the overall desire of providing predictability and capacity planning does not address such a determination with regard to an individual new request that is received. That is, when considering the context in which the predictability and capacity planning are described in *Jackson*, one of ordinary skill in the art would not understand those terms are describing a determination of whether acceptance of a new request will violate availability of a shared resource at some point in the future. Instead, as discussed above, *Jackson's* technique expressly teaches accepting or rejecting requests based solely on whether the bandwidth of the system is sufficient for acceptance of the request at the time it is received, without any consideration of whether the acceptance of the request might lead to future violations.

That is, *Jackson* fails to teach an admission controller that is operable to determine whether acceptance of a new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services. The general statements regarding predictability and capacity planning in *Jackson* in no way refer to determining whether acceptance of an individual, new request will result in such a violation at any point in the future. That is, the statements in *Jackson* regarding predictability and capacity planning are provided in an overview context, and are not made with respect to consideration of an individual, new request that is received. Instead, as discussed above, *Jackson* appears to determine whether a bandwidth limit assigned for a given web site being hosted is exceeded, and if it is exceeded, then an additional request for service by such web site is rejected. While this appears to mention rejecting the acceptance of additional requests once the defined bandwidth limit for a given web site is exceeded, this fails to teach determining whether acceptance of a new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services. Rather, *Jackson* appears to propose accepting new requests until a defined bandwidth is exceeded, without any consideration

of whether acceptance of a given request will violate availability of a desired amount of usage of shared resources by a hosting service at some point in the future.

In view of the above, Appellant respectfully requests that the rejection of claim 11 be overturned.

Claims 12-13 and 16 each depend either directly or indirectly from independent claim 11, and are thus likewise believed to be allowable at least based on their dependency from claim 11 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 12-13 and 16 also be overturned.

Dependent Claim 14

Dependent claim 14 depends from claim 11, and thus inherits all of the limitations of claim 11 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 14 is allowable at least because of its dependence from claim 11 for the reasons discussed above.

Further, claim 14 recites "wherein the admission controller is operable to use a segment-based model of the memory resources to determine whether at least a portion of a requested streaming file is in the memory resources." *Jackson* fails to teach this further element of claim 14. As discussed above with claim 1, *Jackson* fails to teach a segment-based memory model. Thus, *Jackson* fails to teach an admission controller that uses such a segment-based memory model to determine whether at least a portion of a requested streaming file is in the memory resources.

Thus, for this further reason, the rejection of claim 14 should also be overturned.

Independent Claim 18 and Dependent Claims 19-21

Independent claim 18 recites:

> A method for managing admission of requests to hosting services that share resources, the method comprising:
>
> allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files to clients thereof, wherein said set of shared resources comprises memory;
>
> for each of the plurality of hosting services, identifying a desired amount of usage of the set of shared resources to be available for the hosting service; and
>
> isolating usage of the set of shared resources by the plurality of hosting services to ensure that the respective desired amount of usage of the set of shared resources is available to each hosting service, wherein said isolating usage of the set of shared resources comprises:
>
> specifying, for each of the hosting services, an amount of usage of the set of shared resources to be available, at any time, to the hosting service; and
>
> determining whether acceptance of a new request for service by a hosting service will violate, at any point in the future, availability of a specified amount of usage of the shared resources for any of the plurality of hosting services. (Emphasis added).

Jackson fails to teach at least the above-emphasized element of claim 18. That is, as discussed above with claim 11, *Jackson* fails to teach determining whether acceptance of a new request for service by a hosting service will violate, at any point in the future, availability of a specified amount of usage of the shared resources for any of the plurality of hosting services.

In view of the above, Appellant respectfully requests that the rejection of claim 18 be overturned.

Claims 19-21 each depend either directly or indirectly from independent claim 18, and are thus likewise believed to be allowable at least based on their dependency from claim 18 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 19-21 also be overturned.

Independent Claim 23 and Dependent Claims 24-26

Independent claim 23 recites:

> A method for managing admission of requests to a hosting service, the method comprising:
> allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files to clients thereof;
> for each of the hosting services, identifying a desired amount of usage of the set of shared resources to be available for the hosting service;
> receiving a new request for a streaming file to be served by one of the hosting services;
> determining, based at least in part on a modeled memory state of the shared resources, a cost to the one of the hosting services for serving the requested streaming file, wherein the cost corresponds to the shared resources to be consumed in serving the requested streaming file; and
> determining, based at least in part on the cost, whether to admit the new request for service by the one of the hosting services. (Emphasis added).

Jackson fails to teach at least the above-emphasized elements of claim 23. For instance, *Jackson* fails to teach a modeled memory state of the shared resources (as discussed above with claim 1), and thus fails to teach determining a cost to a hosting service for serving a requested streaming file based at least in part on such a modeled memory state. Thus, for at least this reason, Appellant respectfully requests that the rejection of claim 23 be overturned.

Claims 24-26 each depend either directly or indirectly from independent claim 23, and are thus likewise believed to be allowable at least based on their dependency from claim 23 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 24-26 also be overturned.

Dependent Claim 27

Dependent claim 27 depends from claim 23, and thus inherits all of the limitations of claim 23 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 27 is allowable at least because of its dependence from claim 23 for the reasons discussed above.

Claim 27 further recites "wherein said determining whether to admit the new request comprises: determining whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request; and determining whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services." (emphasis added). As discussed above with claim 11, *Jackson* fails to teach making such a determination of whether acceptance of a new request will violate availability of a desired amount of usage of shared resources at any point in the future. Rather, *Jackson* teaches a technique in which new requests are accepted until a defined bandwidth is exceeded, without any consideration of whether acceptance of a given request will violate availability of a desired amount of usage of shared resources by a hosting service at some point in the future.

Thus, for this further reason, the rejection of claim 27 should also be overturned.

Dependent Claim 39

Dependent claim 39 depends from claim 23, and thus inherits all of the limitations of claim 23 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 39 is allowable at least because of its dependence from claim 23 for the reasons discussed above.

Claim 39 further recites "wherein said determining said cost, based at least in part on a modeled memory state of the shared resources, comprises: determining, based at least in part on a segment-based model of memory of the shared resources" (emphasis added). As discussed above with claim 1, *Jackson* fails to teach any such segment-based model of memory.

Thus, for this further reason, the rejection of claim 39 should also be overturned.

Independent Claim 28 and Dependent Claim 29

Independent claim 28 recites:

> A method comprising:
> allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files to clients thereof, wherein the shared resources includes a memory;
> receiving, at a time T_{cur}, a new request for a streaming file to be served by one of the hosting services;
> creating a segment-based model of the memory as of time T_{cur}; and
> based at least in part on the segment-based model of the memory, determining whether to accept the received request for service by the hosting service. (Emphasis added).

Jackson fails to teach at least the above-emphasized elements of claim 28. As discussed above with claim 1, *Jackson* fails to teach a segment-based model of memory. As such, *Jackson* also fails to teach determining, based at least in part on such a segment-based model of memory, whether to accept a received request for service by a hosting service. Thus, for at least this reason, Appellant respectfully requests that the rejection of claim 28 be overturned.

Claim 29 depends from independent claim 28, and is thus likewise believed to be allowable at least based on its dependency from claim 28 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claim 29 also be overturned.

Dependent Claim 30

Dependent claim 30 depends from claim 28, and thus inherits all of the limitations of claim 28 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 30 is allowable at least because of its dependence from claim 28 for the reasons discussed above.

Claim 30 further recites "wherein said determining whether to accept the received request for service by the hosting service comprises: determining whether the requested hosting service has sufficient available resource usage allocated thereto to service the received request; and determining whether acceptance of the received request for service by the requested hosting service will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services." (emphasis added). As discussed above with claim 11, *Jackson* fails to teach making such a determination of whether acceptance of a new request will violate availability of a desired amount of usage of shared resources at any point in the future. Rather, *Jackson* teaches a technique in which new requests are accepted until a defined bandwidth is exceeded, without any consideration of whether acceptance of a given request will violate availability of a desired amount of usage of shared resources by a hosting service at some point in the future.

Thus, for this further reason, the rejection of claim 30 should also be overturned.

Dependent Claim 31

Dependent claim 31 depends from claim 28, and thus inherits all of the limitations of claim 28 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 31 is allowable at least because of its dependence from claim 28 for the reasons discussed above.

Claim 31 further recites "wherein said segment-based model of the memory comprises (a) identification of unique segments of streaming files previously accessed by clients and (b) identification of corresponding timestamps of most recent accesses of each unique segment" (emphasis added). As discussed above with claim 1, *Jackson* fails to teach a segment-based model of memory, and certainly fails to teach any such model that comprises the elements further recited by claim 31.

Thus, for this further reason, the rejection of claim 31 should also be overturned.

Independent Claim 32 and Dependent Claim 35

Independent claim 32 recites:

> Software code stored to a computer-readable medium, which when executed causes a computer to perform a method comprising:
> creating a segment-based model of a media server's memory, wherein the media server's memory is a shared resource to which a plurality of hosting services implemented on the media server have access for serving their respective files to clients thereof; and
> determining whether to serve a requested streaming file from one of the plurality of hosting services based at least in part on the segment-based model of the media server's memory. (Emphasis added).

Jackson fails to teach at least the above-emphasized elements of claim 32. As discussed above with claim 1, *Jackson* fails to teach a segment-based model of memory. As such, *Jackson* also fails to teach determining, based at least in part on such a segment-based model of memory, whether to serve a requested streaming file from one of the plurality of hosting services. Thus, for at least this reason, Appellant respectfully requests that the rejection of claim 32 be overturned.

Claim 35 depends from independent claim 32, and is thus likewise believed to be allowable at least based on its dependency from claim 32 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claim 35 also be overturned.

Dependent Claim 33

Dependent claim 33 depends from claim 32, and thus inherits all of the limitations of claim 32 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 33 is allowable at least because of its dependence from claim 32 for the reasons discussed above.

Claim 33 further recites "determining whether the one of the plurality of hosting services has sufficient available resource usage allocated thereto to serve the requested streaming file; and determining whether acceptance of the received request for service by the one of the plurality of

hosting services will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services" (emphasis added). As discussed above with claim 11, *Jackson* fails to teach making such a determination of whether acceptance of a new request will violate availability of a desired amount of usage of shared resources at any point in the future. Rather, *Jackson* teaches a technique in which new requests are accepted until a defined bandwidth is exceeded, without any consideration of whether acceptance of a given request will violate availability of a desired amount of usage of shared resources by a hosting service at some point in the future.

Thus, for this further reason, the rejection of claim 33 should also be overturned.

Dependent Claim 34

Dependent claim 34 depends from claim 32, and thus inherits all of the limitations of claim 32 in addition to its own supplied limitations. It is respectfully submitted that dependent claim 34 is allowable at least because of its dependence from claim 32 for the reasons discussed above.

Claim 34 further recites "wherein said segment-based model of the media server's memory comprises (a) identification of unique segments of streaming files previously accessed by clients of the media server and (b) identification of corresponding timestamps of most recent accesses of each unique segment" (emphasis added). As discussed above with claims 1 and 32, *Jackson* fails to teach a segment-based model of memory, and certainly fails to teach any such model that comprises the elements further recited by claim 34.

Thus, for this further reason, the rejection of claim 34 should also be overturned.

Independent Claim 36 and Dependent Claims 37-38

Independent claim 36 recites:

> An admission controller for managing admission of requests to hosting services that share resources, the admission controller comprising:
> means for receiving a new request for a streaming file to be served by one of a plurality of hosting services that share access to a set of shared resources for serving their respective files to clients thereof;
> means for performing a resource availability check for the one of a plurality of hosting services from which the streaming file is requested by the new request to determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request; and
> means for performing performance isolation guarantee check for the plurality of hosting services to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services. (Emphasis added).

Jackson fails to teach at least the above-emphasized element of claim 36. That is, as discussed above with claim 11, *Jackson* fails to teach determining whether acceptance of a new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

In view of the above, Appellant respectfully requests that the rejection of claim 36 be overturned.

Claims 37-38 each depend either directly or indirectly from independent claim 36, and are thus likewise believed to be allowable at least based on their dependency from claim 36 for the reasons discussed above. Accordingly, Appellant respectfully requests that the rejection of claims 37-38 also be overturned.

Conclusion

In view of the above, Appellant requests that the board overturn the outstanding rejections of claims 1-4, 6-14, 16, 18-21, and 23-39. Attached hereto are a Claims Appendix, Evidence Appendix, and Related Proceedings Appendix. As noted in the attached Evidence Appendix, no evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the Examiner is being submitted. Also, as noted by the Related Proceedings Appendix, no related proceedings are referenced in II above, and thus no copies of decisions in related proceedings are provided.

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: September 28, 2007

Signature: Donna Forbit
(Donna Forbit)

Respectfully submitted,

By
Jody C. Bishop
Attorney/ Agent for Applicant(s)
Registration No.: 44,034
Dated: September 28, 2007
Telephone No. (214) 855-8007

VIII. CLAIMS APPENDIX

Claims Involved in the Appeal of Application Serial No. 10/601,357

1. A method for managing admission of requests to a shared media server, the method comprising:

allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective streaming files to clients, wherein said set of shared resources comprises memory; and

managing admission of client requests for streaming files to each of the plurality of hosting services to ensure that a desired amount of usage of the shared resources is available to each hosting service, wherein said managing admission of client requests for streaming files comprises:

receiving a client request for a streaming file to be served from one of said hosting services; and

using a segment-based memory model to determine whether at least a portion of the requested streaming file is in the memory.

2. The method of claim 1 further comprising:

implementing the plurality of hosting services on a shared media server.

3. The method of claim 1 wherein the set of shared resources comprises:

shared memory resources and shared disk resources.

4. The method of claim 1 further comprising:

determining the desired amount of usage of the shared resources for a hosting service from a service level agreement.

5. (Canceled)

6. The method of claim 1 further comprising:

determining from the segment-based memory model a cost associated with the one of said hosting services serving the requested streaming file.

7. The method of claim 1 wherein said managing admission of client requests for streaming files comprises:

receiving a new request for service of a streaming file by one of the plurality of hosting services;

performing a resource availability check for the one of a plurality of hosting services to determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request.

8. The method of claim 7 wherein said managing admission of client requests for streaming files further comprises:

performing a performance isolation guarantee check for the plurality of hosting services to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

9. The method of claim 1 further comprising:

specifying, for each of the plurality of hosting services, a desired amount of usage of the shared resources to be available at any given time for the hosting service.

10. The method of claim 9 wherein said managing admission of client requests comprises:

managing admission of client requests for streaming files to each of the plurality of hosting services to ensure that each of the plurality of hosting services has usage of its corresponding specified desired amount of the shared resources.

11. A system comprising:

a media server comprising a plurality of hosting services for streaming files implemented thereon, wherein the media server comprises shared resources and wherein the plurality of hosting services share usage of the media server's shared resources in serving streaming files to their respective clients; and

an admission controller for managing admission of client requests for service to each of the plurality of hosting services to ensure that no one of the plurality of hosting services

overtakes usage of an undesirably high proportion of the shared resources;

wherein said admission controller is operable to receive a new request for service of a streaming file by one of the plurality of hosting services, and determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request; and

wherein said admission controller is further operable to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

12. The system of claim 11 wherein the admission controller is operable to manage admission of said client requests to the plurality of hosting services to ensure that a desired amount of usage of the shared resources is available, at any given time, to each hosting service.

13. The system of claim 11 wherein the shared resources comprise memory resources and disk resources.

14. The system of claim 13 wherein the admission controller is operable to use a segment-based model of the memory resources to determine whether at least a portion of a requested streaming file is in the memory resources.

15. (Canceled)

16. The system of claim 11 wherein an amount of resource usage is preallocated to the requested hosting service.

17. (Canceled)

18. A method for managing admission of requests to hosting services that share resources, the method comprising:

allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files to clients thereof, wherein said set of shared resources comprises memory;

for each of the plurality of hosting services, identifying a desired amount of usage of the set of shared resources to be available for the hosting service; and

isolating usage of the set of shared resources by the plurality of hosting services to ensure that the respective desired amount of usage of the set of shared resources is available to each hosting service, wherein said isolating usage of the set of shared resources comprises:

specifying, for each of the hosting services, an amount of usage of the set of shared resources to be available, at any time, to the hosting service; and

determining whether acceptance of a new request for service by a hosting service will violate, at any point in the future, availability of a specified amount of usage of the shared resources for any of the plurality of hosting services.

19. The method of claim 18 wherein said hosting services host streaming files for access by clients thereof.

20. The method of claim 18 wherein the set of shared resources comprise:

shared memory resources and shared disk resources.

21. The method of claim 18 further comprising:

determining the desired amount of usage of the set of shared resources for a hosting service from a service level agreement.

22. (Canceled)

23. A method for managing admission of requests to a hosting service, the method comprising:

allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files to clients thereof;

for each of the hosting services, identifying a desired amount of usage of the set of shared resources to be available for the hosting service;

receiving a new request for a streaming file to be served by one of the hosting services;

determining, based at least in part on a modeled memory state of the shared resources, a cost to the one of the hosting services for serving the requested streaming file, wherein the cost corresponds to the shared resources to be consumed in serving the requested streaming file; and

determining, based at least in part on the cost, whether to admit the new request for service by the one of the hosting services.

24. The method of claim 23 wherein said determining whether to admit the new request comprises:

determining whether the cost exceeds the amount of usage of the shared resources allowed for the one of the hosting services.

25. The method of claim 23 wherein said determining whether to admit the new request comprises:

determining whether the cost of shared resources consumed violates availability of a desired amount of usage of the set of shared resources to be available for another one of the hosting services.

26. The method of claim 23 wherein the set of shared resources comprises:

shared memory resources and shared disk resources.

27. The method of claim 23 wherein said determining whether to admit the new request comprises:

determining whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request; and

determining whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

28. A method comprising:

allowing each of a plurality of hosting services access to any of a set of shared resources for serving their respective files to clients thereof, wherein the shared resources includes a memory;

receiving, at a time T_{cur}, a new request for a streaming file to be served by one of the hosting services;

creating a segment-based model of the memory as of time T_{cur}; and

based at least in part on the segment-based model of the memory, determining whether to accept the received request for service by the hosting service.

29. The method of claim 28 further comprising:

for each of the plurality of hosting services, identifying a desired amount of usage of the set of shared resources to be available at any time for the hosting service.

30. The method of claim 28 wherein said determining whether to accept the received request for service by the hosting service comprises:

determining whether the requested hosting service has sufficient available resource usage allocated thereto to service the received request; and

determining whether acceptance of the received request for service by the requested hosting service will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

31. The method of claim 28 wherein said segment-based model of the memory comprises (a) identification of unique segments of streaming files previously accessed by clients

and (b) identification of corresponding timestamps of most recent accesses of each unique segment.

32. Software code stored to a computer-readable medium, which when executed causes a computer to perform a method comprising:

creating a segment-based model of a media server's memory, wherein the media server's memory is a shared resource to which a plurality of hosting services implemented on the media server have access for serving their respective files to clients thereof; and

determining whether to serve a requested streaming file from one of the plurality of hosting services based at least in part on the segment-based model of the media server's memory.

33. The software code of claim 32 wherein said code for determining whether to serve a requested streaming file from one of the plurality of hosting services comprises code, which when executed causes a computer to perform a method comprising:

determining whether the one of the plurality of hosting services has sufficient available resource usage allocated thereto to serve the requested streaming file; and

determining whether acceptance of the received request for service by the one of the plurality of hosting services will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

34. The computer-executable software code of claim 32 wherein said segment-based model of the media server's memory comprises (a) identification of unique segments of streaming files previously accessed by clients of the media server and (b) identification of corresponding timestamps of most recent accesses of each unique segment.

35. The computer-executable software code of claim 32 wherein said code for determining whether to serve a requested streaming file from one of the plurality of hosting services comprises code, which when executed causes a computer to perform a method comprising:

determining a cost to the one of the plurality of hosting services for serving the requested streaming file, wherein the cost corresponds to the amount of the shared resources to be consumed in serving the requested streaming file.

36. An admission controller for managing admission of requests to hosting services that share resources, the admission controller comprising:

means for receiving a new request for a streaming file to be served by one of a plurality of hosting services that share access to a set of shared resources for serving their respective files to clients thereof;

means for performing a resource availability check for the one of a plurality of hosting services from which the streaming file is requested by the new request to determine whether the requested hosting service has sufficient available resource usage allocated thereto to service the new request; and

means for performing performance isolation guarantee check for the plurality of hosting services to determine whether acceptance of the new request will violate, at any point in the future, availability of a desired amount of usage of the shared resources for any of the plurality of hosting services.

37. The admission controller of claim 36 wherein said means for performing a resource availability check comprises:

means for determining a cost associated with the one of a plurality of hosting services serving the requested streaming media file, wherein the cost corresponds to the shared resources to be consumed in serving the requested streaming file.

38. The admission controller of claim 36 wherein said set of shared resources comprises:

memory and disk resources.

39. The method of claim 23 wherein said determining said cost, based at least in part on a modeled memory state of the shared resources, comprises:

determining, based at least in part on a segment-based model of memory of the shared resources.

IX. EVIDENCE APPENDIX

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS APPENDIX

Appellant respectfully notes that the following co-pending application is on appeal before the Board, which contains at least some issues that are similar to issues of the present application, which may be affected or have a bearing on the Board's decision in this appeal. An Appeal Brief was filed for Application No. 10/601,992 on July 19, 2007. No decision has been received in this related appeal, and thus no copy of such decision is provided.

There are no other related proceedings.